Module5\_Homework.R

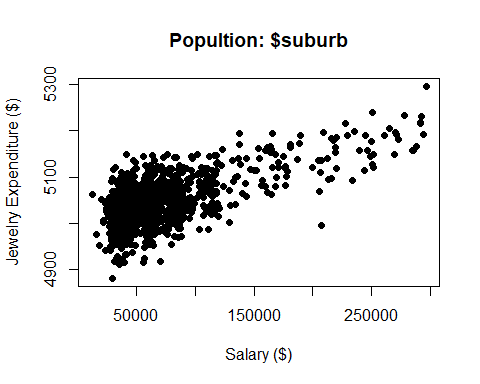
mmhan\_uricwmy

2022-08-08

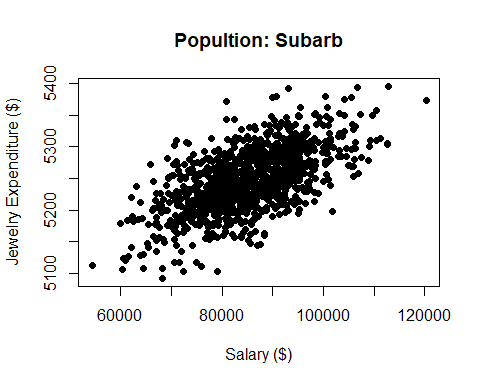
# =====================================================  
# Module 5 Practice - Bivariate Linear Regression  
# Supervised Modeling Technique  
# Mike Hankinson  
# October 26, 2021  
# =====================================================  
  
# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
# Givens:   
 # Data Set with values of yearly salary (thousands) and money spent of jewelry (thousands) for   
 # two samples from two different populations:   
 # i. Individuals from the city   
 # ii. Individuals from the suburbs.   
  
# Unknowns:   
 # Perform independent bivariate linear regressions for each sample to predict jewelry spend   
 # using yearly salary.   
  
# Determine:  
 # 1. City Regression:  
 # a. Report and interpret the slope coefficient for salary.  
 # b. Report and interpret the intercept coefficient for salary.   
 # Does this value make sense to you?  
 # c. Report the t-statistic and p-value for salary.   
 # Is salary a significant predictor of jewelry spend in this sample?  
 # d. Report and interpret the R-squared value.  
 #   
 # 2. Suburb Regression  
 # a. Report and interpret the slope coefficient for salary.  
 # b. Report and interpret the intercept coefficient for salary.   
 # Does this value make sense to you?  
 # c. Report the t-statistic and p-value for salary.   
 # Is salary a significant predictor of jewelry spend in this sample?  
 # d. Report and interpret the R-squared value.  
 #   
 # 3. Compare and contrast the effect of salary of jewelry spend in these   
 # two different populations. Can you explain why we see a difference   
 # in the slope coefficients in the two regressions?  
  
# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
# Process  
# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
# 1. Load and Plot Both Sets of Data  
# 2. City Regression:  
# a. Perform Regression / Obtain Summary (coefficients, T-stat, P-Value and R^2)  
# b. Plot Regression Line.  
# c. Verify Linearity of Model (against defining assumptions)  
# i. Normality of Residuals  
# \* Plot Histogram of Residuals  
# \* Plot Residuals against Predicted Jewelry Expenditure  
# \* Normal Q-Q Plot of Residual Values  
# ii. Homoscedasticity - Constant Variance  
# d. Determine Confidence Level for B1  
# e. Evaluate Model Stability: Repeatedly Sample Data (Create 1,000 train/test splits)  
# Do-Loop:  
# i. Create train/test sample  
# ii. Perform linear regression model on training data  
# iii. Save B1 coefficient  
# iv. Calculate R^2 for test sample as squared correlation between Y from test and  
# predicted values for test sample (Y^hat). Calculate % decrease.  
# Post Do-Loop:  
# v. Histogram and Summary of drop in R^2  
# vi. Evaluate B1 parameter / Build 95% Confidence Level  
# vii. Conclusion  
# 3. Suburb Regression:  
# a. Perform Regression / Obtain Summary (coefficients, T-stat, P-Value and R^2)  
# b. Plot Regression Line.  
# c. Verify Linearity of Model (against defining assumptions)  
# i. Normality of Residuals  
# \* Plot Histogram of Residuals  
# \* Plot Residuals against Predicted Jewelry Expenditure  
# \* Normal Q-Q Plot of Residual Values  
# ii. Homoscedasticity - Constant Variance  
# d. Determine Confidence Level for B1  
# e. Evaluate Model Stability: Repeatedly Sample Data (Create 1,000 train/test splits)  
# Do-Loop:  
# i. Create train/test sample  
# ii. Perform linear regression model on training data  
# iii. Save B1 coefficient  
# iv. Calculate R^2 for test sample as squared correlation between Y from test and  
# predicted values for test sample (Y^hat). Calculate % decrease.  
# Post Do-Loop:  
# v. Histogram and Summary of drop in R^2  
# vi. Evaluate B1 parameter / Build 95% Confidence Level  
# vii. Conclusion  
# 4. Model Comparison: City vs. Suburb

# 1. Load and Plot Both Sets of Data

city <- read.csv("Assignment5City.csv")  
suburb <- read.csv("Assignment5Suburb.csv")  
  
  
plot(city$Salary, city$Jewelry, pch=16, ylab="Jewelry Expenditure ($)",   
 xlab="Salary ($)", main="Popultion: $suburb" ) #plot(x,y)



plot(suburb$Salary, suburb$Jewelry, pch=16, ylab="Jewelry Expenditure ($)",   
 xlab="Salary ($)", main="Popultion: Subarb")



# 2. $suburb Regression

# a. Perform Regression / Obtain Summary (coefficients, T-stat, P-Value and R^2)  
# ..........................................  
city.mod <- lm(Jewelry ~ Salary, city) #lm(y~x)  
  
summary(city.mod) # B0 = $4,999 ;B1 = 0.0007 (slope coefficient for X -- Jewelry Expenditure/Salary)

##   
## Call:  
## lm(formula = Jewelry ~ Salary, data = city)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -149.404 -28.534 -0.313 29.723 120.738   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 4.999e+03 2.075e+00 2409.07 <2e-16 \*\*\*  
## Salary 7.010e-04 2.517e-05 27.85 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 40.53 on 1198 degrees of freedom  
## Multiple R-squared: 0.3931, Adjusted R-squared: 0.3926   
## F-statistic: 775.9 on 1 and 1198 DF, p-value: < 2.2e-16

# Call:  
 # lm(formula = Jewelry ~ Salary, data = city)  
 #   
 # Residuals:  
 # Min 1Q Median 3Q Max   
 # -149.404 -28.534 -0.313 29.723 120.738   
 #   
 # Coefficients:  
 # Estimate Std. Error t value Pr(>|t|)   
 # (Intercept) 4.999e+03 2.075e+00 2409.07 <2e-16 \*\*\*  
 # Salary 7.010e-04 2.517e-05 27.85 <2e-16 \*\*\*  
 # ---  
 # Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
 #   
 # Residual standard error: 40.53 on 1198 degrees of freedom  
 # Multiple R-squared: 0.3931, Adjusted R-squared: 0.3926   
 # F-statistic: 775.9 on 1 and 1198 DF, p-value: < 2.2e-16  
  
  
library(broom)  
betas.city <- city.mod$coefficients  
print(betas.city)

## (Intercept) Salary   
## 4.998825e+03 7.009760e-04

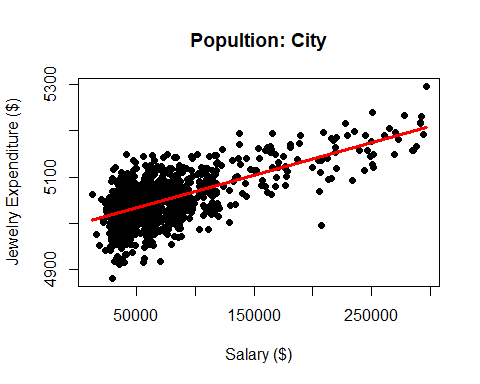
coef(summary(city.mod))[2, "t value"] # Salary t-statistic

## [1] 27.85424

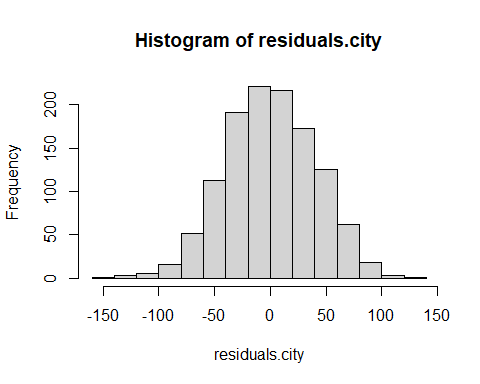
glance(city.mod)$p.value # Salary p-value

## value   
## 4.636129e-132

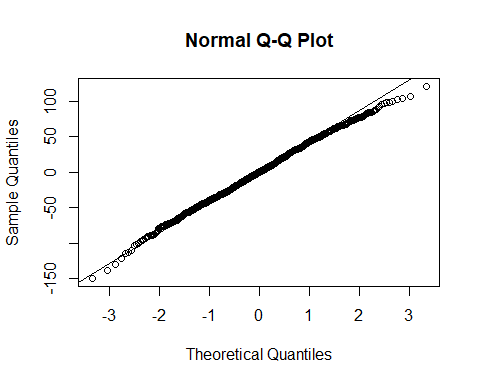
# Required Data Summary:  
 # B0 $4999   
 # B1 Salary 0.000700976 ($ 0.0007 spent on Jewelry per dollar earned)  
 # Salary t-statistic 27.85424   
 # Salary p-value < 2.2e-16 (4.636129e-132)  
 # R^2 0.3931  
  
 # - The y-intercept of the model (B0, at 0 salary) shows a minimum of $4,998.   
 # spent on jewelry annually.   
 # - The slope coefficient of 0.00070 states that the population spent an additional  
 # $ 0.0007 on Jewelry per $1 earned.  
 # - The p-value is < 0.001 so the null hypothesis is rejected   
 # or, test is statistically significant -- there is correlation between salary  
 # and jewelry expenditure in City populations  
 # - However, an R^2 value shows that 39.3% of the variability in jewelry spend can  
 # be attributed to salary.   
 # - We will continue to evaluate the validity of the model by verifying its linearity,  
 # determining the confidence level of B1 and evaluating its stability through   
 # repeated sampling.   
  
  
# b. Plot Regression Line  
# ..........................................  
  
plot(city$Salary, city$Jewelry, pch=16, ylab="Jewelry Expenditure ($)",   
 xlab="Salary ($)", main="Popultion: City")  
mins.range <-floor(min(city$Salary)):ceiling(max(city$Salary))  
lines(mins.range, betas.city[1]+mins.range\*betas.city[2], col="red", lwd=3)



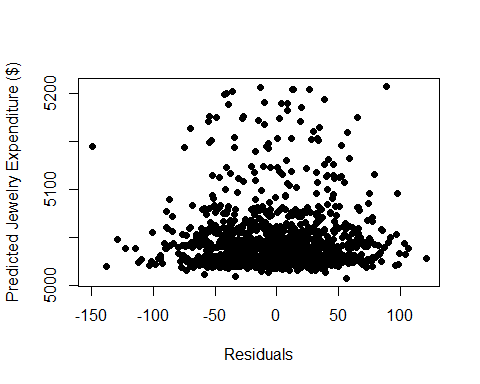
# Appear to have found a good fit, but need to analyze the residuals to be certain.   
 # To check if the residuals are normally distributed, and our model complies with   
 # the assumption of normality, let's look at a histogram of the residual values:  
  
  
# c. Verify Linearity of Model (against defining assumptions)  
# ..........................................  
  
# i. Normality of Residuals: Plot Histogram of Residuals and Q-Q Plot  
# ..........................................  
  
# \*Plot Histogram of Residuals  
residuals.city <- city.mod$residuals  
hist(residuals.city)



# The distribution of the residuals looks to have a close to normal distribution --   
 # as desired.   
  
# \* Q-Q plot of the residual values  
qqnorm(residuals.city)  
qqline(residuals.city)



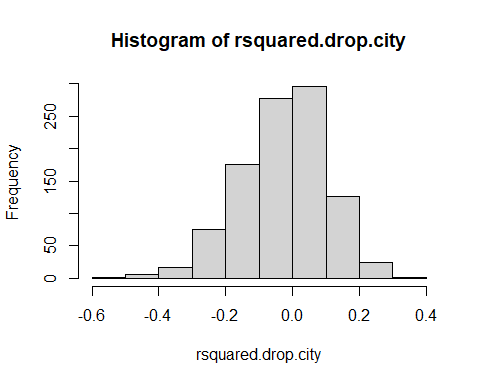
# - The Q-Q plot shows the theoretical line the residuals should follow   
 # if normally distributed.   
 # - The actual residuals are overlaid, as well.   
 # - The residuals begin to slightly deviate from the theoretical distribution at 1.5 on the x-axis,   
 # The desired output us no deviations from-2 to 2.  
 # - However, since there are only 2 features within this data set, we cannot attempt to   
 # divide the data into further subsets (to look for linearity within the subsets).  
  
  
# ii. Homoscedasticity - Constant Variance of Residuals  
# ..........................................  
# Reminder about Residuals:  
 # - the difference between the OBSERVED value and the MEAN VALUE the   
 # model predicts for THAT SPECIFIC VALUE.   
 # - It indicates the extent to which a model accounts for the variation within the   
 # observed data.   
 # - The variance of the error term should not depend on X  
  
  
# \* Plot Residuals against Predicted Jewelry Expenditure  
plot(city.mod$residuals, city.mod$fitted.values, pch=16, xlab="Residuals", ylab="Predicted Jewelry Expenditure ($)")



# - The plot shows that as Y increases, the range of possible error   
 # term values remains nearly constant across the x-axis.   
 # - This then meets the Homoscedasticity criteria.   
  
  
# d. Determine Confidence Level for B1  
# ..........................................  
# In a standard normal distribution with a mean = 0 and sd = 1, we expect 95%   
# of realizations to fall between -1.96 and 1.96. In other words, 95% of the time,   
# a normally distributed variable should be within a 1.96 standard deviation of the mean.  
# Leverage this fact to create a confidence interval for B1:  
# C.I. = B1 +- 1.96\*se(B1)  
# The correct interpretation for this confidence interval is that for 95% of samples,   
# the true value of B1 will be contained in this interval.   
  
confint(city.mod)

## 2.5 % 97.5 %  
## (Intercept) 4.994754e+03 5.002896e+03  
## Salary 6.516019e-04 7.503500e-04

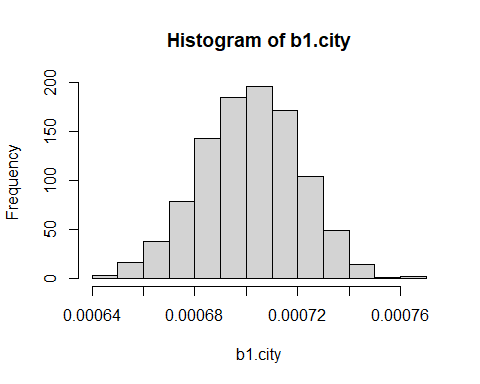
# 2.5 % 97.5 %  
 # (Intercept) 49,947 50,029 B0 Min.Jewelry Expenditure, $  
 # Salary 0.00065 0.00075 B1 Jewelry Expenditure/Salary  
  
  
  
# e. Evaluate Model Stability: Repeatedly Sample Data (Create 1,000 train/test splits)  
# ..........................................  
# Repeatedly sample the data, creating 1,000 different train/test splits.   
# For each split, save the estimated B1 coefficient, and record the decrease   
# in R squared between our train and test samples to evaluate the model stability.  
  
  
# Steps i-iv - Do-Loop  
# ..........................................  
  
b1.city <- rsquared.drop.city <- numeric (1000)  
for(i in 1:1000){  
 train.rows.city <- sample(1:1000, 650)  
 train.dat.city <- city[train.rows.city,]  
 test.dat.city <- city[-train.rows.city,] # All rows BUT train rows.  
 lin.city.mod <- lm(Jewelry ~ Salary, data=train.dat.city)   
 b1.city[i] <- lin.city.mod$coefficients[2]  
 r.train.city <- summary(lin.city.mod)$r.squared  
 r.test.city <- cor(test.dat.city$Jewelry, predict(lin.city.mod, newdata=test.dat.city))^2   
 rsquared.drop.city[i] <- (r.train.city-r.test.city)/r.train.city  
}  
  
# v. Histogram and Summary of drop in R^2  
# ..........................................  
hist(rsquared.drop.city)



summary(rsquared.drop.city)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -0.50857 -0.10997 -0.01789 -0.02765 0.06364 0.32524

# Min. 1st Qu. Median Mean 3rd Qu. Max.   
 # -0.45442 -0.11125 -0.01495 -0.02983 0.05874 0.30366   
  
 # Drop in R^2 being is close to to 0%,  
 # Sometimes, the test model outperforms the trained model.   
  
  
  
# vi. Evaluate B1 parameter / Build 95% Confidence Level  
# ..........................................  
# Evaluate the B1 parameter, and build a 95% confidence interval using the mean and   
# standard deviations from the collection of 1,000 possible B1 values.  
hist(b1.city)



# Repeated Sample Confidence Interval  
lower.bound.city <- mean(b1.city) + qt(0.025, df=98)\*sd(b1.city) #bottom 2.5%; qt() - Student t Distribution  
upper.bound.city <- mean(b1.city) + qt(0.975, df=98)\*sd(b1.city) #top 2.5%  
  
lower.bound.city # [1] 0.0006613527

## [1] 0.0006630676

upper.bound.city # [1] 0.0007412432

## [1] 0.0007400899

# Compare,  
# Model result of B1 vs avg. of 1,000   
cbind(Full\_Model.city=betas.city[2], Repeated\_Sample\_city=mean(b1.city))

## Full\_Model.city Repeated\_Sample\_city  
## Salary 0.000700976 0.0007015787

# Full\_Model.city Repeated\_Sample\_city  
 # Salary 0.000700976 0.0007012979  
  
  
rbind(Full\_Model.city=confint(city.mod)[2,], Repeated\_sample\_city=c(lower.bound.city, upper.bound.city))

## 2.5 % 97.5 %  
## Full\_Model.city 0.0006516019 0.0007503500  
## Repeated\_sample\_city 0.0006630676 0.0007400899

# 2.5 % 97.5 %  
 # Full\_Model.city 0.000651 0.000750 B1 Jewelry Expenditure/Salary  
 # Repeated\_sample\_city 0.000663 0.000741 B1 Jewelry Expenditure/Salary  
  
  
# vii. Conclusions  
# ..........................................  
 # - B1 follows a normal distribution wIth repeated sampling.   
 # - The full model estimate of B1 is nearly identical to B1 calculated from  
 # the mean of the 1,000 B1 parameters sampled. but that the confidence interval has become tighter.   
 # - The confidence interval of the continued sampling B1 is tighter than that of the  
 # full model.   
 # - After repeatedly sampling, the true value of B1 is known to a narrower range.

# 3. Suburb Regression

# a. Perform Regression / Obtain Summary  
# ..........................................  
suburb.mod <- lm(Jewelry ~ Salary, suburb) #lm(y~x)  
  
summary(suburb.mod) # B0 = $4,983 ;B1 = 0.0031 (slope coefficient for X -- Jewelry Expenditure/Salary)

##   
## Call:  
## lm(formula = Jewelry ~ Salary, data = suburb)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -125.533 -26.529 0.196 25.502 138.066   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 4.983e+03 1.092e+01 456.45 <2e-16 \*\*\*  
## Salary 3.105e-03 1.276e-04 24.34 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 39.64 on 998 degrees of freedom  
## Multiple R-squared: 0.3726, Adjusted R-squared: 0.3719   
## F-statistic: 592.6 on 1 and 998 DF, p-value: < 2.2e-16

# Call:  
 # lm(formula = Jewelry ~ Salary, data = suburb)  
 #   
 # Residuals:  
 # Min 1Q Median 3Q Max   
 # -125.533 -26.529 0.196 25.502 138.066   
 #   
 # Coefficients:  
 # Estimate Std. Error t value Pr(>|t|)   
 # (Intercept) 4.983e+03 1.092e+01 456.45 <2e-16 \*\*\*  
 # Salary 3.105e-03 1.276e-04 24.34 <2e-16 \*\*\*  
 # ---  
 # Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
 #   
 # Residual standard error: 39.64 on 998 degrees of freedom  
 # Multiple R-squared: 0.3726, Adjusted R-squared: 0.3719   
 # F-statistic: 592.6 on 1 and 998 DF, p-value: < 2.2e-16   
  
  
  
library(broom)  
betas.suburb <- suburb.mod$coefficients  
print(betas.suburb)

## (Intercept) Salary   
## 4.983096e+03 3.105394e-03

coef(summary(suburb.mod))[2, "t value"] # Salary t-statistic

## [1] 24.34363

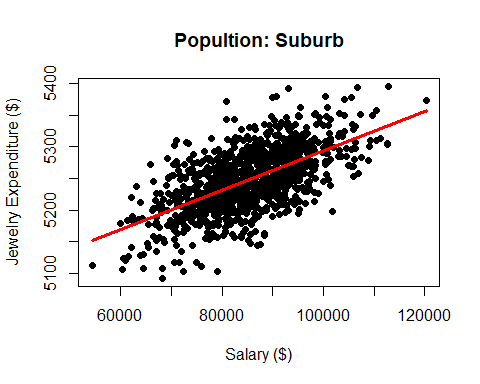
glance(suburb.mod)$p.value # Salary p-value

## value   
## 3.994508e-103

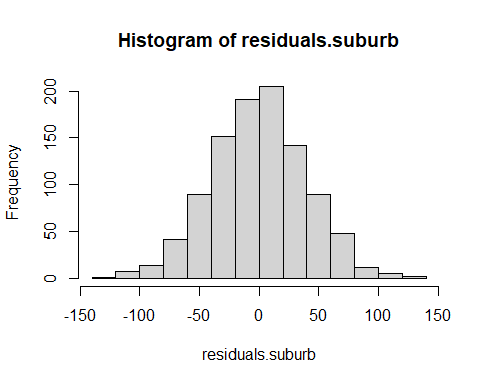
# Required Data Summary:  
 # B0 $4,983.  
 # B1 Salary 0.00310  
 # Salary t-statistic 24.34363   
 # Salary p-value < 2.2e-16 (3.994508e-103)  
 # R^2 0.3726  
   
   
 # - The y-intercept of the model (B0, at 0 salary) shows a minimum of $4,983   
 # spent on jewelry annually.   
 # - The slope coefficient of 0.0031 states that the population spent an additional  
 # $ 0.0031 on Jewelry per $1 earned.  
 # - The p-value is < 0.001 so the null hypothesis is rejected   
 # or, test is statistically significant -- there is correlation between salary  
 # and jewelry expenditure in City populations  
 # - However, an R^2 value shows that 39.6% of the variability in jewelry spend can  
 # be attributed to salary.   
 # - We will continue to evaluate the validity of the model by verifying its linearity,  
 # determining the confidence level of B1 and evaluating its stability through   
 # repeated sampling.   
  
  
  
# b. Plot Regression Line  
# ..........................................  
betas.suburb <- suburb.mod$coefficients  
print(betas.suburb)

## (Intercept) Salary   
## 4.983096e+03 3.105394e-03

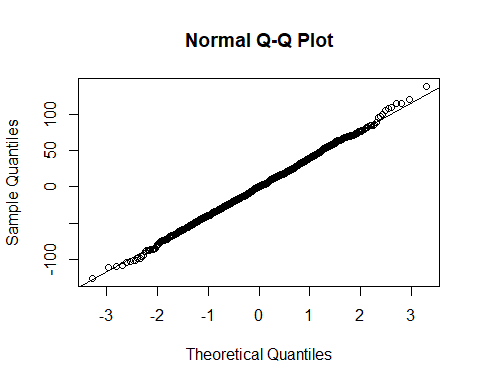
# (Intercept) Salary   
 # 4.983096e+03 3.105394e-03   
 # Jewelry Expenditure @ 0 Jewelry Expenditure/Salary  
  
  
plot(suburb$Salary, suburb$Jewelry, pch=16, ylab="Jewelry Expenditure ($)",   
 xlab="Salary ($)", main="Popultion: Suburb")  
mins.range <-floor(min(suburb$Salary)):ceiling(max(suburb$Salary))  
lines(mins.range, betas.suburb[1]+mins.range\*betas.suburb[2], col="red", lwd=3)



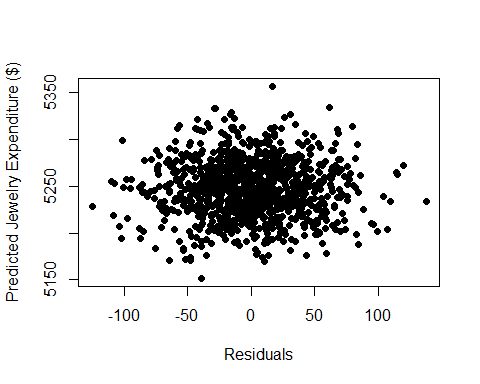
# Appear to have found a good fit, but need to analyze the residuals to be certain.   
 # To check if the residuals are normally distributed, and our model complies with   
 # the assumption of normality, let's look at a histogram of the residual values:  
  
  
# c. Verify Linearity of Model (against defining assumptions)  
# ..........................................  
  
# i. Normality of Residuals: Plot Histogram of Residuals and Q-Q Plot  
# ..........................................  
  
# \*Plot Histogram of Residuals  
residuals.suburb <- suburb.mod$residuals  
hist(residuals.suburb)



# The distribution of the residuals appears fairly normal. Maybe a little left side heavy  
  
# \* Q-Q plot of the residual values  
qqnorm(residuals.suburb)  
qqline(residuals.suburb)



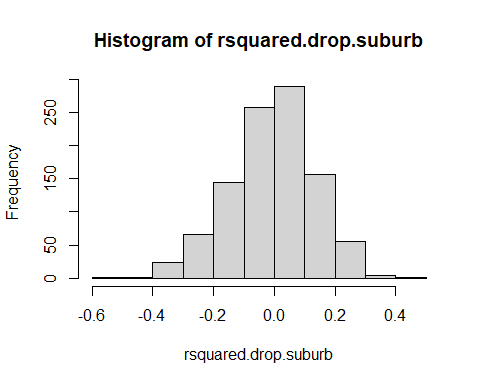
# - The Q-Q plot shows the theoretical line the residuals should follow   
 # if normally distributed.   
 # - The actual residuals are overlaid, as well.  
 # - The residuals don't deviate from the theoretical distribution in the range on the x-axis,   
 # from -2 to 2, which demonstrates normaility of the residuals.  
  
  
# ii. Homoscedasticity - Constant Variance of Residuals  
# ..........................................  
# Reminder about Residuals:  
 # - the difference between the OBSERVED value and the MEAN VALUE the   
 # model predicts for THAT SPECIFIC VALUE.   
 # - It indicates the extent to which a model accounts for the variation within the   
 # observed data.   
 # - The variance of the error term should not depend on X  
  
  
# \* Plot Residuals against Predicted Jewelry Expenditure  
plot(suburb.mod$residuals, suburb.mod$fitted.values, pch=16, xlab="Residuals", ylab="Predicted Jewelry Expenditure ($)")



# - The plot shows that as Y increases, the range of possible error   
 # term values remains nearly constant across the x-axis.   
 # - This then meets the Homoscedasticity criteria.   
  
  
# d. Determine Confidence Level for B1  
# ..........................................  
# In a standard normal distribution with a mean = 0 and sd = 1, we expect 95%   
# of realizations to fall between -1.96 and 1.96. In other words, 95% of the time,   
# a normally distributed variable should be within a 1.96 standard deviation of the mean.  
# Leverage this fact to create a confidence interval for B1:  
# C.I. = B1 +- 1.96\*se(B1)  
# The correct interpretation for this confidence interval is that for 95% of samples,   
# the true value of B1 will be contained in this interval.   
  
confint(suburb.mod)

## 2.5 % 97.5 %  
## (Intercept) 4.961673e+03 5.004519e+03  
## Salary 2.855067e-03 3.355720e-03

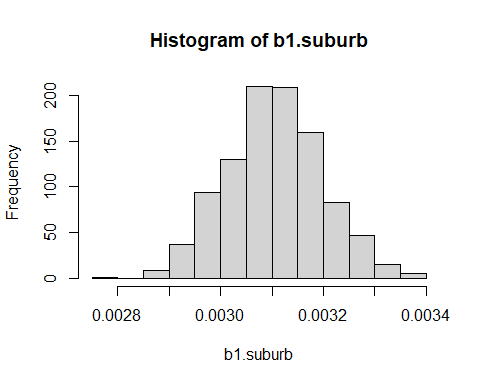
# 2.5 % 97.5 %  
# (Intercept) 4961.672567 5004.51878 B0 Min. Jewelry Expenditure  
# Salary 0.002855 0.00335 B1 Jewelry Expenditure/Salary  
  
  
  
# e. Evaluate Model Stability: Repeatedly Sample Data (Create 1,000 train/test splits)  
# ..........................................  
# Repeatedly sample the data, creating 1,000 different train/test splits.   
# For each split, save the estimated B1 coefficient, and record the decrease   
# in R squared between our train and test samples to evaluate the model stability.  
  
  
# Steps i-iv - Do-Loop  
# ..........................................  
  
b1.suburb <- rsquared.drop.suburb <- numeric (1000)  
for(i in 1:1000){  
 train.rows.suburb <- sample(1:1000, 650)  
 train.dat.suburb <- suburb[train.rows.suburb,]  
 test.dat.suburb <- suburb[-train.rows.suburb,] # All rows BUT train rows.  
 lin.suburb.mod <- lm(Jewelry ~ Salary, data=train.dat.suburb)   
 b1.suburb[i] <- lin.suburb.mod$coefficients[2]  
 r.train.suburb <- summary(lin.suburb.mod)$r.squared  
 r.test.suburb <- cor(test.dat.suburb$Jewelry, predict(lin.suburb.mod, newdata=test.dat.suburb))^2   
 rsquared.drop.suburb[i] <- (r.train.suburb-r.test.suburb)/r.train.suburb  
}  
  
# v. Histogram and Summary of drop in R^2  
# ..........................................  
hist(rsquared.drop.suburb)



summary(rsquared.drop.suburb)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -0.575120 -0.094205 0.001328 -0.006176 0.085486 0.440050

# Min. 1st Qu. Median Mean 3rd Qu. Max.   
 # -0.499345 -0.090576 0.003509 -0.003699 0.090816 0.513282   
  
 # Drop in R^2 being is close to to 0%,  
 # Sometimes, the test model outperforms the trained model.   
  
  
  
# vi. Evaluate B1 parameter / Build 95% Confidence Level  
# ..........................................  
# Evaluate the B1 parameter, and build a 95% confidence interval using the mean and   
# standard deviations from the collection of 1,000 possible B1 values.  
hist(b1.suburb)



# Repeated Sample Confidence Interval  
lower.bound.suburb <- mean(b1.suburb) + qt(0.025, df=98)\*sd(b1.suburb) #bottom 2.5%; qt() - Student t Distribution  
upper.bound.suburb <- mean(b1.suburb) + qt(0.975, df=98)\*sd(b1.suburb) #top 2.5%  
  
lower.bound.suburb # [1] 0.002934469

## [1] 0.002919042

upper.bound.suburb # [1] 0.003275835

## [1] 0.003291099

# Compare,  
# Model result of B1 vs avg. of 1,000   
cbind(Full\_Model.suburb=betas.suburb[2], Repeated\_Sample\_suburb=mean(b1.suburb))

## Full\_Model.suburb Repeated\_Sample\_suburb  
## Salary 0.003105394 0.003105071

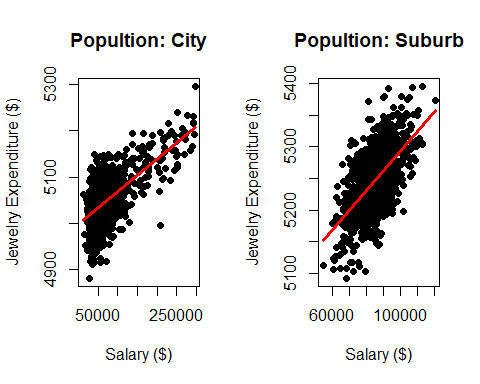
# Full\_Model.suburb Repeated\_Sample\_suburb  
 # Salary 0.003105394 0.003105358  
  
  
rbind(Full\_Model.suburb=confint(suburb.mod)[2,], Repeated\_sample\_suburb=c(lower.bound.suburb, upper.bound.suburb))

## 2.5 % 97.5 %  
## Full\_Model.suburb 0.002855067 0.003355720  
## Repeated\_sample\_suburb 0.002919042 0.003291099

# 2.5 % 97.5 %  
 # Full\_Model.suburb 0.002855067 0.003355720 B1 Jewelry Expenditure/Salary  
 # Repeated\_sample\_suburb 0.002921949 0.003288767 B1 Jewelry Expenditure/Salary  
  
  
  
# vii. Conclusions  
# ..........................................  
 # - In terms of sampling, SUBURB results followed same TREND as did CITY.   
 # - B1 follows a normal distribution wIth repeated sampling.   
 # - The full model estimate of B1 is nearly identical to B1 calculated from  
 # the mean of the 1,000 B1 parameters sampled. but that the confidence interval has become tighter.   
 # - The confidence interval of the continued sampling B1 is tighter than that of the  
 # full model.   
 # - After repeatedly sampling, the true value of B1 is known to a narrower range.

# 4. Model Comparison: City vs. Suburb

# DATA SUMMARY: CITY SUBURB  
# .......................................................................  
# B0 $4999. $4,983.  
# B1 Salary 0.000700 0.003105  
# Salary t-statistic 27.85424 24.34363  
# Salary p-value < 2.2e-16 < 2.2e-16   
# R^2 0.3931 0.3726  
# Salary Full\_Model 0.000700 0.003105   
# Repeated\_Sample 0.000701 0.003105  
# Full\_Model 2.5% 0.000651 0.002855   
# Full\_Model 97.5% 0.000750 0.003355   
# Repeated\_sample 2.5% 0.000663 0.002921   
# Repeated\_sample 97.5% 0.000741 0.003288  
  
  
  
# The data for both city and suburb populations is similar.   
# - Baseline spend (y-intercept) is only $16 different between the 2 populations.   
# - Both groups reject the null hypothesis. There is a definitive relationship  
# between salary and jewelry spend.   
# - Both models' jewelry spend variability can only slightly be attributed to   
# salary (39% City, 37% suburb).  
# - Those that live in the suburb spend much more than those that live in   
# the city above the base of B0....  
# \* City B1 at $0.70 per $1,000 earned   
# \* Suburb B1 at $3.10 per $1,000 earned   
  
  
  
#grid  
par(mfrow=c(1,2))  
#plot1  
plot(city$Salary, city$Jewelry, pch=16, ylab="Jewelry Expenditure ($)",   
 xlab="Salary ($)", main="Popultion: City")  
mins.range <-floor(min(city$Salary)):ceiling(max(city$Salary))  
lines(mins.range, betas.city[1]+mins.range\*betas.city[2], col="red", lwd=3)  
  
#plot2  
plot(suburb$Salary, suburb$Jewelry, pch=16, ylab="Jewelry Expenditure ($)",   
 xlab="Salary ($)", main="Popultion: Suburb")  
mins.range <-floor(min(suburb$Salary)):ceiling(max(suburb$Salary))  
lines(mins.range, betas.suburb[1]+mins.range\*betas.suburb[2], col="red", lwd=3)



# Functions Used

# lm() Fitting Linear Models  
  
# summary() summary is a generic function used to produce result summaries   
# of the results of various model fitting functions.   
# The function invokes particular methods which depend on the class   
# of the first argument.  
  
# floor() floor takes a single numeric argument x and returns a   
# numeric vector containing the largest integers not   
# greater than the corresponding elements of x.   
  
# ceiling() takes a single numeric argument x and returns a numeric vector   
# containing the smallest integers not less than the corresponding   
# elements of x  
  
# c() Combine Values into a Vector or List  
  
# predict() generic function for predictions from the results of various   
# model fitting functions. The function invokes particular methods   
# which depend on the class of the first argument.  
#  
# newdata = within predict(), An optional data frame in which to look for   
# variables with which to predict.   
# If omitted, the fitted values are used.  
  
# confint() Confidence Intervals for Model Parameters  
# sample() Random Samples.....sample(x, size)  
# qt() Student t Distribution  
  
# par() Set or Query Graphical Parameters.  
# set by specifying them as arguments to par in tag = value form,   
# or by passing them as a list of tagged values.  
  
# which() Give the TRUE indices of a logical object, allowing for array indices.  
  
  
  
  
# rm(list = ls()) Removes global environment